A Method of Personalized Web Search Result Clustering Based on Formal Concept Analysis

Jing Wang Yajun Du

School of Mathematics and Computer Engineering, Xihua University, Chengdu 610039, China

Abstract

Most existing Web search result clustering techniques, generally anchoring in pure content-based analysis, generate a single set of clusters for all individuals without tailoring to individuals' preferences and thus are unable to support personalization. In this paper, we incorporate a target user's categorization preferences into the Web search result clustering process using Formal Concept Analysis (FCA). Personalized conceptual clusters hierarchy of Web search result will be built combining content analysis and user information analysis. We focus on the target user's categorization preference extracting and cluster hierarchy building based on FCA.

Keywords: Web search result clustering, Personalized document clustering, Formal Concept Analysis (FCA), Search engine.

1. Introduction

Both the number of users and the amount of information available have exploded since the advent of the World Wide Web (WWW). Most of Web users use various search engines such as Yahoo [19] and Google [17] to get specific information. A key factor in the success of Web search engines is their ability to rapidly find good quality results to queries that are based on rather specific terms, like ''technology of planting rose''. On the other side, however, traditional search services usually fall short when asked to answer much broader queries, for example, to find documents about the term ''program''. The poor quality of results in these cases is mainly due to three different factors: (1) the user's "intention behind the search" is not clearly expressed by too general, short queries; (2) the search terms are polysemous or synonymous; (3) the number of results returned to the user is excessively high. One approach that tries to solve this problem is using clustering techniques for grouping similar document together in order to facilitate presentation of results in more compact form and enable thematic browsing of the results set. Hearst and Pedersen [7] showed that relevant documents tend to be more similar to each other, thus the clustering of similar search results helps users find relevant results.

Most traditional clustering algorithms cannot be directly used for search result clustering. Zamir and Etzioni [10] gave a good analysis on these issues and identified some key requirements for search result clustering: (1) the clustering algorithm should group similar documents together to generate coherent clusters; (2) the generated clusters should have readable descriptions for quick browsing by users; (3) the clustering algorithm should generate overlapping clusters as documents often have multiple topics; (4) the clustering algorithm should be fast enough for online calculation. In fact, the idea of clustering search results as a means to improve retrieval performance has been investigated quite deeply in Information Retrieval. A seminal work in this respect is the Scatter/Gather project [7]. Scatter-Gather provides a simple graphical user interface to do clustering on a traditional information retrieval system. Grouper [10] was the first publicly available software to address the search result clustering problem. The main feature of Grouper is the introduction of a phrase-analysis algorithm called STC (Suffix Tree Clustering). Grouper has inspired a number of other proposals along the same lines. For example, Lingo/Carrot Search [13] extend the STC algorithm with the use of SVD (Singular Value Decomposition) in order to improve the quality of the produced clusters. In addition, various industrial systems implement Web search result clustering in their (meta-) search engines such as Vivisimo [18].

Since Wille [11] developed Formal Concept Analysis (FCA), concept lattice, the core data structure in FCA, has been used widely in machine learning, data mining and knowledge discovery, information retrieval, etc. [16]. FCA is a conceptual clustering technique which has some advantages over standard document clustering algorithms for clustering Web search result: (1) FCA provides an intrinsic description of each cluster, which makes clusters more interpretable; (2) intent and extent of formal concept are uniform which insures grouping similar documents together to generate coherent clusters; (3) concept lattice reflects the relation of all of the concepts which makes the implement of overlapping clusters is more easy; (4) cluster is organized as a lattice which facilitate recovery from bad decisions while exploring the hierarchy and, in general, provides a richer and more flexible way of browsing the document space. For theses advantages, some of Web search results clustering systems applying FCA have been presented such as CREDO [3] and JBraindead [9].

At present, general Web search result clustering techniques have been anchored in pure content-based analysis. As a consequence, most existing Web search result clustering techniques are not tailored to individuals' preferences and therefore are unable to facilitate personalization. Given the same query terms, they presented the same cluster result to different users. However, the target user's categorization preferences are usually personal, and, a user's Web document search typically is guided by his or her categorization scheme. For example, given a set of research articles related to "data mining," researchers engaged in developing novel data mining techniques may prefer organizing the articles according to underlying techniques (e.g., classification analysis, clustering analysis, association rules and sequential patterns). In contrast, researchers who are applying data mining techniques to solve business questions generally would prefer categories based on application domains (e.g., banking, manufacturing, health care and telecommunications). Hence, effective Web search result clustering should consider individual preferences and needs to support personalization in Web document categorization.

In this paper, we proposed a personalized method of Web search result clustering based on FCA. We incorporate a target user's categorization preferences into the document-clustering process using FCA. Personalized concept lattice clusters hierarchy of Web search result was built combining content analysis and user information analysis. The users with different categorization preferences will be provided different concept lattice clusters results on line. But as known of us, FCA is computationally more costly than standard clustering, lattices generated by FCA can be big, complex and hence difficult to use for practical browsing purposes. Hence, in order to reduce the time of FCA computing and make the cluster result is easy to be browsed by users we construct a partial concept lattice layer by layer rather than a full concept lattice to build the personalized conceptual cluster hierarchy of Web search result.

2. Formal Concept Analysis

We recall the basics of Formal Concept Analysis (FCA) as far as they are needed for this paper.

To allow a mathematical description of concepts as being composed of extensions and intensions, Formal Concept Analysis starts with a formal context. A formal context is a triple $K := (G, T, I)$, where G is a set of objects, T is a set of attributes, and I is a binary relation between G and T (i.e. $I \subseteq G \times T$). If object $g \in G$ has attribute $t \in T$ then g is related I to t which is indicated by the relationship $(g, t) \in I$.

From a formal context, a concept hierarchy, called concept lattice, can be derived. For $X \subseteq G$, we define $X' := \{\mathrm{t} \in \mathrm{T} \mid \forall \mathrm{g} \in \mathrm{X}; (\mathrm{g}, \mathrm{t}) \in \mathrm{I}\}$ and, for $Y \subseteq T$,we define $Y' := \{\mathrm{g} \in \mathrm{G} \mid \forall \mathrm{t} \in \mathrm{Y}; (\mathrm{g}, \mathrm{t}) \in \mathrm{I}\}$.

A formal concept of a formal context (G, T, I) is defined as a pair (X, Y) with $X \subseteq G$, $Y \subseteq T$, $X' = Y$ and $Y' = X$. The sets X and Y are called the extent and intent of the formal concept (X, Y). The subconcept-superconcept relation is formalized by $(X_1, Y_1) \leq (X_2, Y_2) \Leftrightarrow X_1 \subseteq X_2 (Y_1 \supseteq Y_2)$. The set of all formal concepts of a context K together with the partial order $\leq$ is called the concept lattice of K.

3. Personalized clustering Web search result based on FCA

Our personalized method of Web search result clustering based on FCA is mainly composed of the following steps:

(1) Web search result fetching;

(2) The target user's categorization preference extracting;

(3) Cluster hierarchy building using FCA;

(4) The resulting personalized cluster hierarchy presenting.

Given a query Q by the user, we get the web pages returned by a certain Web search engine as the Web search results corpus G. These web pages have been analyzed by an HTML parser and result items are extracted. Generally, there are only titles and query-dependent snippets available in each result item. We assume these contents are informative enough because most search engines are well designed to facilitate users' relevance judgment only by the title and snippet, thus it is able to present the most relevant contents for a given query.

For personalized clustering the Web search results, we extract the target user's categorization preference by collecting, analyzing and picking up topics from the information about the user. All of the web pages the target user visited form the user interest document set. Optimal representative terms describing the set of user interest document are considered as the user interest topic terms set T that reflects the target user's categorization preference.

Then, Formal Concept Analysis is applied to the Web search result corpus G as objects, where the attributes of each document are the subset of the set of user interest topic terms T which are contained in its text. On the formal context, the personalized cluster hierarchy is built through constructing partial concept lattice layer by layer. The resulting personalized cluster hierarchy closed to the target user's categorization preference is presented to be browsed for the user.

For producing personalized concept lattices with better clustering features, the key steps of our approach lies on the target user's categorization preference extracting and cluster hierarchy building using FCA. In the following, we will focus on both of them.

3.1. Extraction of the user's categorization preference

The target user's categorization preference is reflected by the topics he or she interested in. In other words, these topics contained in the target user's interest express categorization scheme he or she prefers to. So, in order to capture the target user's categorization preference, systems need collect information about the user, analyze and pick up topics from the information.

Information can be colleted from users in two ways: explicitly, for example asking for feedback such as preferences or ratings; and implicitly, for example observing user behaviors such as the time spent reading an online document. Explicit collecting of user interests has several drawbacks. The user provides inconsistent or incorrect information, the user's interests may change over time, and it places a burden on the user that they may not wish to accept. Thus, many research efforts are underway to implicitly collect user interests [2][8][5]. User browsing histories are the most frequently used source of information about user interests. The fact that a user has visited a page is an indication of user interest in that page's content. Based on this idea, we collect all of the web pages the target user visited as the user interest document set P. Collecting user interests in such implicit way can provide privacy protection and avoid additional effort requires for the user.

Then, we extract the target user's categorization preference by picking up topics from the user interest document set. Optimal representative terms describing the set of user interest document are considered as user interest topic terms. For selecting such optimal representative terms capable of reflecting the target user's categorization preference, we use a variation of terminological weight formula introduced in [1]. A terminological weight is designed to find, in a collection which is representative from some specific domain, terms which are more suitable as descriptors for the domain. A terminological weight compares the domain-specific collection with a collection from a different domain, and assigns a higher weight to terms that are more frequent in the domain-specific collection than in the contrastive collection. In our case, the domain-specific collection can be the user interest document set P; and the contrastive collection is the Web search result corpus G minus the user interest document set P:

$$\mathrm{w}_{\mathrm{i}} = 1 - \frac{1}{\log_2\left(2 + \dfrac{\mathrm{tf}_{\mathrm{i,ret}} \cdot \mathrm{f}_{\mathrm{i,ret}} - 1}{\mathrm{tf}_{\mathrm{i,col}} + 1}\right)}$$

where wi is the terminological weight of term i, tf,i,ret represents the relative frequency of term i in the user interest document set P, fi,ret is the user interest document set P document frequency of term i, and tfi,col is the relative frequency of term i in the Web search result corpus G minus the user interest document set P.

From the user interest document set, the first n terms ranked according to the value of the terminological weight are selected as the user interest topic terms set T. They are the optimal topic descriptions of the documents in the domain that the user interested in. And, they reflect the target user's categorization preference in a way.

3.2. Personalized cluster hierarchy building using FCA

The formal context $(G, T_{\text{preference}}, I)$ of Web search result in our application is formed through the set of Web search result corpus G as formal objects, the topic terms set $T_{\text{preference}}$ that reflect the target user's categorization preference as formal attributes and a binary relation I between G and $T_{\text{preference}}$. $I:(g,t)\in I$ indicates that the document $g \in G$ contains term $t \in T_{\text{preference}}$. The personalized cluster hierarchy is built by constructing partial concept lattice on the formal context $(G, T_{\text{preference}}, I)$.

In the applications of FCA, building concept lattice efficiently is an important task, for which various algorithms [12][6][15] have been developed. Recently, Xie Run etc. proposed an algorithm of hierarchic construction of concept lattice, for short, HCCL [14]. They defined that the layer serial number of a formal concept (X, Y) of the concept lattice is equal to the length of maximum chain from (X, Y) to the top concept of the concept lattice, i.e., if the length of maximum chain from a formal concept (X, Y) of the concept lattice to the top concept of the concept lattice is N then the formal concept (X, Y) just lies on the N layer.

Based on this hierarchy structure of concept lattice, two important properties of hierarchical concept lattice were presented by the authors: the concepts in a same layer are incomparable and a concept is overlaid by at least one concept on the upper layer. The general formula for the object and attribute mapping was derived, and a theorem, which describes the structural invariability of objects during their construction, was obtained. According to the properties and theorem the algorithm of HCCL was proposed by the authors. HCCL generates concept lattice layer by layer based on the hierarchical structure of concept lattice. No abundant concept is created, because filtration is carried out during construction of concepts.

We improve on the algorithm of HCCL to build the personalized Web search cluster hierarchy in our approach. Found on the same hierarchy structure of concept lattice as HCCL we construct the top H layer of concept lattice to build the cluster hierarchy. Considering three factors: (1) it is terribly time-consuming to construct a full concept lattice; (2) complicated concept lattice is difficult for quick browsing by users; (3) the topics of those concepts that lie on the higher layers is more narrow so it is not significant to display narrow-topic concept as cluster to users, we construct the top H layer of concept lattice rather than a full concept lattice to build the cluster hierarchy.

The top element of personalized cluster hierarchy is generated through query terms Q as the description of it and Web search result corpus G as the documents of it. Clearly, the pair of (G, Q) is not always a formal concept. However, it represents the query topic of the user and so we consider it as a dummy concept. Personalized cluster hierarchy is seen as a set of concepts C and of a set of edges E, where the edges are ordered pairs of concepts (C_1, C_2) such that $C_1 \longrightarrow C_2$, i.e., C_1 is a lower neighbor of C_2. Similar to CREDO, all the documents of one concept that are not covered by its children are grouped in a dummy concept named “other”.

Personalized cluster hierarchy will be generated layer by layer until the number of current layer is up to “HmaxNum” which is the limitation of the cluster hierarchy. Two function of “FindNextCandidate” and “FindNextTrue” is created based on the idea of the algorithm HCCL. “FindNextCandidate” generates candidate clusters of the next layer and “FindNextTrue” finds the true clusters from these candidate clusters. In addition, we improve HCCL for our application by establishing exact ordered relations between the neighbor two layers since it is needed for our application.

The pseudo-code of our cluster hierarchy building algorithm is shown as follows.

BuildClusterHierarchy

Input: Query terms Q, hierarchy limitation HmaxNum, formal context (G, $T_{preference}$, I)

Output: The personalized cluster hierarchy CH = (C, E)

```
1.  C := {(G, Q)}  /* Generate the top element */
2.  Ccurrent := {G, $\varnothing$ }
3.  C1:= FindNextCandidate ((G, $T_{preference}$, I), Ccurrent)
4.  C := C $\bigcup$ C1
5.  for each (X, Y) $\in$ C1
6.    add edge (X,Y) $\longrightarrow$ (G, Q) to E
7.  end for
8.  Ccurrent := C1
9.  Cnext := $\varnothing$
10. HnextNum := 2
/* The following statements generate clusters layer by layer and cluster hierarchy is limited by HmaxNum */
11. while HnextNum <= HmaxNum
12.   CnextCandidate := FindNextCandidate ((G, $T_{preference}$, I ), Ccurrent)
13.   Cnext := FindNextTrue (CnextCandidate )
14.   C := C $\bigcup$ Cnext
/* Line 15 to 26 establish exact ordered relations between the neighbor layer */
15.   for each (X, Y) $\in$ Ccurrent
16.     DocuChild := $\varnothing$
17.     for each (Xe, Ye) $\in$ Cnext
18.       if Y $\subset$ Ye then
19.         add edge (Xe,Ye) $\longrightarrow$ (X, Y) to E
20.         DocuChild := DocuChild $\bigcup$ Xe
21.       endif
22.     end for
23.     if DocuChild = $\varnothing$ then
24.       break
25.     end
26.   end for
27.   Ccurrent := Cnext
28.   HnextNum := HnextNum +1
29. end while
30. return (C, E)
```

/* Generate candidate clusters of the next layer */

FindNextCandidate((G, $T_{preference}$, I), Ccurrent)

```
1.  CnextCandidate := $\varnothing$
2.  YattributeSet := $\varnothing$
3.  for each (X, Y) $\in$ Ccurrent
4.    Jcandidate := $\varnothing$
5.    DAttributeSet := $\varnothing$
6.    M0 := T $-$ ( YattributeSet $\bigcup$ Y )
7.    M := M0
8.    while M $\neq$ $\varnothing$
9.      ASet := $\varnothing$
10.     for each m $\in$ M
11.       A := (Y $\bigcup$ {m})'
12.       if A $\neq$ $\varnothing$ then
13.         Aset := Aset $\bigcup$ {A}
14.       end if
15.     end for
16.     if ASet = $\varnothing$ then
17.       break
18.     end if
19.     MaxNum := max {|A| | A $\in$ ASet}
20.     for each A $\in$ ASet
21.       if |A| = MaxNum then
22.         D := $\bigcap_{a \in A} ( a')$
23.         DAttributeSet := DAttributeSet $\bigcup$ D
24.         if M $\neq$ M0 then
25.           if $\forall$ (Xc, Yc) $\in$ Jcandidate, A $\not\subset$ Xc then
26.           Jandidate := Jcandidate $\bigcup$ { (A, D) }
27.             end if
28.           else
29.             Jcandidate := Jcandidate $\bigcup$ { (A, D) }
30.           end if
31.         end if
32.     end for
33.     M := T $-$ (YattributeSet $\bigcup$ Y $\bigcup$ DattributeSet)
34.   end while
```

```
  YattributeSet := YattributeSet $\bigcup$ Y
  CnextCandidate := CnextCandidate $\bigcup$ Jcandidate
end for
return CnextCandidate

/* Find the true clusters from the candidate clusters */
FindNextTrue（CnextCandidate）
Cnext := $\varnothing$
CSortCandidate := Sort ( CnextCandidate)
MaxNum := max $\{|E| \mid (E, F) \in$ CSortCandidate $\}$
for each $(X, Y) \in$ CSortCandidate
  if $|X|$ = MaxNum then
    add (X, Y) to Cnext
  else
    if $\forall (Xe,Ye) \in$ Cnext, $X \not\subset Xe$ then
      add (X, Y) to Cnext
    end if
  end if
end for
return Cnext
```

4. Conclusions

Web search result clustering is an efficient approach of improving retrieval performance of Web search engine. Usually, a user's Web document search typically is guided by his or her categorization scheme. Effective Web search result clustering should consider individual preferences and needs to support personalization in Web document categorization. Most existing Web search result clustering techniques, generally anchoring in pure content-based analysis, generate a single set of clusters for all individuals without tailoring to individuals' preferences and thus are unable to support personalization.

In this paper, we present a new method of personalized Web search result clustering based on FCA. The users with different categorization preferences will be provided different concept clusters result on line. The target user's categorization preferences are captured by implicitly collecting, analyzing and extracted topics from the information about the user. We construct a partial concept lattice layer by layer rather than a full concept lattice to build the personalized conceptual cluster hierarchy of Web search result, which can reduce the time of FCA computing and make the cluster result easy to be browsed by users.

In the future, we will apply our approach to build a personalized Web search result clustering system to prove its efficiency ulteriorly.

Acknowledgement

The work is supported by the Education Department Foundation of Sichuan Province (Grant No.2006A086), the Application Foundation of Sichuan Province (Grant No.2006J13-056), the Cultivating Foundation of Scie-nce and Technology of Xihua University (Grant No.R0622611), the Cultivating Foundation of the Science and Technology Leader of Sichuan Province.

5. References

[1] A. Penas etc., Corpus-Based Terminology Extraction applied to Information Access, *Proceedings of Corpus Linguistics*, 2001.

[2] C.C. Chen, M.C. Chen, Y. Sun. PVA, a self-adaptive personal view agent, *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 257-262, 2001.

[3] C. Carpineto and G. Romano, Exploiting the Potential of Concept Lattices for Information Retrieval with CREDO, *Journal of Universal Computer Science*, 10 (8): 985-1013, 2004. http://credo.fub.it.

[4] Chih-Ping Wei, Chin-Sheng Yang, Han-Wei Hsiao, A Collaborative Filtering–Based Approach to Personalized Document Clustering, *DecisionSupport Systems*, 5(8), 2007.

[5] H.R. Kim, P.K. Chan, Learning implicit user interest hierarchy for context in personalization, *Proceedings of the 8th international conference on Intelligent user interfaces*, pp. 101- 108, 2003.

[6] L. Nourine, O. Raynaud, A fast algorithm for building lattices, *Information Processing Letter*, pp. 199-204, 1999.

[7] M.A. Hearst and J.O. Pedersen, Reexamining the Cluster Hypothesis: Scatter/Gather on Retrieval Results, *Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information*, 1996.

[8] M. Claypool , P. Le, M. Waseda, D. Brown. Implicit Interest Indicators, *Proceedings of the 6th international conference on Intelligent user interfaces (ACM)*, pp.33-40, 2001.

[9] M.C. Juan, G. Julio, Anselmo P, etc., Browsing Search Results Via Formal Concept Analysis: Automatic Selection of Attributes, *Proceedings of the Second International Conference on Formal Concept Analysis*, 2004.

[10] O. Zamir and O. Etzioni, Grouper: a dynamic clustering interface for web search results, *Computer Networks*, 31(1) :1361–1374, 1999.

[11] R. Wille, An approach based Restructuring lattice theory: hierarchies of concepts, *Dordrecht-Boston: Reidel*, pp. 445-470, 1982.

[12] R. Godin, R. Missaouf, H. Alaoui. Incremental concept formation algorithms based on Galois (concept) lattice , *Computational Intelligence*, 11(2) :246-267, 1995.

[13] S. Osinski and D. Weiss, A concept-driven algorithm for clustering search results, *IEEE Intelligent Systems*, 20 (3): 48–54, 2005. http://company.carrot-search.com.

[14] R. Xie, H.X. Li, etc., Hierarchic construction of concept lattice, *Jouranal of Southwest Jiaotong University*, 40(6) : 837-841, 2005.

[15] Y.J. Du, Study and implementation on intelligent action of search engine, *Doctor degree*

dissertation, Southwest Jiaotong University, China , 2005.

[16] W.X. Zhang and G.F. Qiu, Uncertain Decision Making Based on Rough Sets, *Tsinghua University Press*, 2005.

[17] http://www.google.com.

[18] http://www.vivisimo.com.

[19] http://www.yahoo.com.